Research Paper | Information Technology | E-ISSN No : 2454-9916 | Volume : 3 | Issue : 4 | Apr 2017

ENHANCING SECURITY USING HONEYWORDS

Vidya Dandinashivara [1] | Shruti Nair [1] | Nikita Paspunatu [1] | Prof. Mrinal Khadse [1]

[1] Department of Information Technology Engineering, SIES Graduate School of Technology, Nerul, Navi Mumbai.

ABSTRACT

Humongous amount of data and information are generated by users everyday which are private, thus needed to be secured and protected. We are using the concept of honeywords to increase the security of the system. Honeywords are basically dummy passwords created by the system which is stored along with the actual user password. When an adversary steals the password file and tries one of the honeywords, the user is immediately informed about this activity.

KEYWORDS: Passwords, Authentication, Honeywords, Security, login.

I. INTRODUCTION:

In today's high technology environment, organizations are becoming more and more dependent on their information systems. The security of the information has always been a challenge. Many systems are been hacked in spite of high end security. Golube showed that the cracking speed of hashes has reached 5.6 billion/s for MD5 and 2.3 billions/s for SHA1 on a single GPU [4]. These advancements make it necessary to develop new security measures. Honeywords are basically fake passwords so as to confuse the attacker with the real and the bogus passwords. These Honeywords can be generated using various techniques such as Password model, Chaffing by tweaking the digits, Hybrid Technique. Generation of Honeywords helps in avoiding the attack of an insider on confidential and important data, thus "Making Data Inconspicuous". Thus the proposed system helps in improving the security of hashed passwords.

II. MOTIVATION:

In the authentication process, where the user has to just register with a password, usually the user may provide a weak password which may be easily cracked by the adversary using either brute force or shoulder-surfing attacks.

In the existing system, if the attacker is successful in stealing the password file, he will get all the details of the user i.e. username and password thus making it easy for the thief to barge in the system. Thus our proposed system produces many dummy passwords and stores it with the original passwords which will confuse the attacker about the authentic password.

III. LITERATURE SURVEY:

Many cases have been registered wherein companies have lost lots of money due to hacked user-password files.Mandiant1's report proved the importance of hacking password files. There was a recent cyber espionage campaign against the New York Times [2]. The recent years have seen innumerable cases of password theft; the secured passwords of Evernote's million users were exposed as were those of users at Yahoo, LinkedIn, and eHarmony, among others [3]. Juels and Rivest provided a new idea of saving the passwords in the file i.e. by storing dummy passwords along with the real passwords [1]. The false passwords are called honeywords. As soon as one of the honeywords is submitted in the login process, the adversary will be detected.

IV. PROPOSED SYSTEM:

In the proposed system, when the user registers, the system fetches the password from the user and produces a set of 34 honeywords which is stored along with the user's actual password. The technique used for the honeyword generation is based on the strength of the user's password.

Table.1 Honeyword Generation Technique used based on the strength of the password.

Strength of the password	Honeyword Generation Technique
Weak	Password model
Moderate	Chaffing by tweaking digits
Strong	Hybrid technique

If the hacker gets access to the password file he will get the entire set of 34 honeywords and also the actual password which are all encrypted. If the adversary tries to login with a honeyword then the actual user password will be changed using either prefix or postfix method. An alert message will be sent to the user about the intrusion along with the changed password.

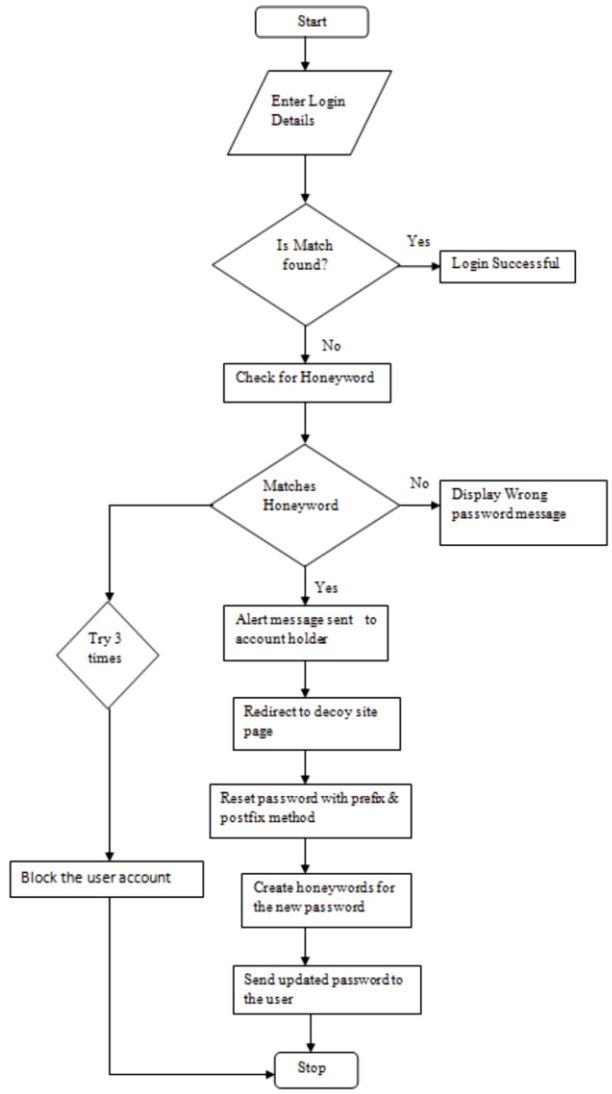

Fig.2 Flowchart of Honeyword Module

OTP Module:

When the user logs into the system successfully he/she has to go through another authentication system which is the OTP module.

In this module the user will again register via android app. The user will be provided with the bank ID in the app. The user will then login to the web application and enter all the user details. The web application will now show the number of SHA256 and MD5 inputs to be entered in order to generate the OTP. When the user enters the same numbers on the android application the OTP is generated. If

it matches with the OTP generated on the web application server then the user has a successful login else an alert message is shown stating "Unsuccessful login".

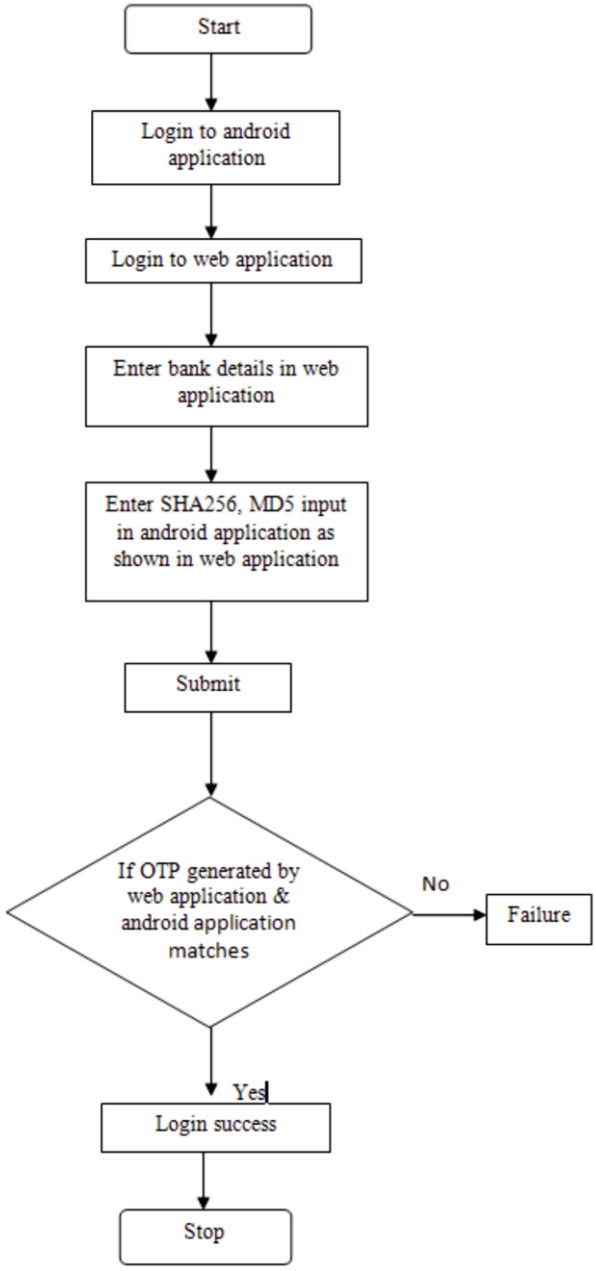

Fig.3 Flowchart of OTP Module

V. WORKING SYSTEM:
The flow of the system is as follows:

1. User registers in the web application i.e. enters the username and password.
2. The honeyword generator produces a set of 34 fake passwords once it fetches the actual password.

These honeywords are produced using various techniques based on the strength of the password as shown in fig.1.The strength of the password is rated according to the score a password receives by the system. This score is decided based on two parameters:

- Length of the password
- Number of numeric and special characters in the password.

Techniques used in producing honeywords:

1. Password Model:
Password model generates honeywords by building a model of real passwords. It retains the number of alphabets, digits and alphanumeric characters without adding any further more characters and randomizes the characters of the password. This model will be used when the strength of the password is low.

Example:
Real password: Ni@#12

Honeywords:
Ax!*63, Qm^!39, Sh*$45, Ue$%25, Tz!&95, Nb!@27, Pu^&41, Ie#$64, Ed(!21, Fs#*96, Yw%$36, Hj(&71, Ax%*32, Qt)!29, Ds$&11, Ov@#64, Ka#!81, No)#55, Rs&%16, Sv!$18, Hm*!29, An*^16, Ms@&18, Dl*$57, Re$#75, Hg^$24, Tr#*65, Uw&)58, Fa!#49, Eq&%37

2. Chaffing by tweaking digits
This method tweaks the digits present in the real password by random numbers and creates honey words

This model will be applied when the strength of the password is moderate.

Example:
Real password: Abc@1995

Honeywords:
Abc@2987, Abc@6478 Abc@4631, Abc@2275, Abc@3691, Abc@1234, Abc@5289, Abc@8912, Abc@3847, Abc@6029, Abc@0106, Abc@3971, Abc@7539, Abc@2589, Abc@4589, Abc@6358, Abc@8534, Abc@1995, Abc@1995, Abc@1995, Abc@2036, Abc@5236, Abc@7864, Abc@6108, Abc@1030, Abc@9880, Abc@4739, Abc@4297, Abc@5495, Abc@2020

3. Hybrid Model.
It is the combination of password model and chaffing by tweaking digits model. This model will be applied when the strength of the password is high.

Example:
Real Password: N!k!t@123!@#$

Honeywords:
Y$u*n$744$$#@, Y$u*n$577$$#@, Y$u*n$845$$#@,
Y$u*n$181$$#@, Y$u*n$195$$#@, Y$u*n$927$$#@,
Y$u*n$872$$#@,
K$v^n@532*^^@, K$v^n@268*^^@,
K$v^n@843*^^@, K$v^n@934*^^@,
K$v^n@937*^^@, K$v^n@201*^^@,
K$v^n@970*^^@,
P+n*q+150^#+@, P+n*q+174^#+@,
P+n*q+259^#+@, P+n*q+835^#+@,
P+n*q+783^#+@, P+n*q+009^#+@,
P+n*q+129^#+@,
M@p+j$784$$++, M@p+j$784$$++,
M@p+j$784$$++,M@p+j$784$$++,
M@p+j$784$$++, M@p+j$784$$++,
M@p+j$784$$++,
S^v&n*121!!##, S^v&n*856!!##,
S^v&n*259!!##, S^v&n*789!!##,
S^v&n*006!!##, S^v&n*169!!##,

OTP module:
For instance if we take the following user details:

- IMEI : 823589069458989
- IMSI : 405151001385048
- ACNO:1337-01
  BankCode:1337
  UserID : 01

Then we follow the procedure as shown below:

1. Concatenate (IMEI, IMSI, ACNO)
2. Concatenate(823589069458989 + 405151001385048 + 133701)
3. Seed = 82358906945898940515100138504813370 1
4. Formula : Seed > HBy(HAx(Seed)) > Final Hash Value > 6 digit OTP
5. It is performed on server & client side in parallel
6. Hashing Function : Used by client & server to generate 6 digit OTP from seed
7. HB = SHA256, HA = MD5
8. Hashing Index (x,y) : Generated by Server, Sent to client
9. x = Rounds for SHA256, y = Rounds for MD5
10. Hashing Index Limit : Min & Max number of rounds to be applied on seed
11. Min : (x,y) = (1,1) rounds, Max : (x,y) = (10,10) rounds

VI. CONCLUSION AND FUTURE SCOPE:

The proposed system is implementation for higher level of authentication using honeywords techniques. When the user registers, the system intelligently decides which technique should be used for honeyword generation. Once unauthorized data access or exposure is suspected we inundate the malicious insider with bogus information in order to dilute the user's real data. The password of user account will be changed and a notification will be sent to the legitimate user. Further to increase the security of the system, OTP module is included. Thus this system ensures high security, where people using the system need not worry about the intrusions or attacks.

REFERENCES:

[1] A. Juels and R. L. Rivest,"Honeywords: Making Password Cracking Detectable", in proceedings o the 2013 ACM SIGSAC Conference on Computer and Communications Security, ser. CCS'13. New York,NY,USA: ACM,2013, pp. 145-160. [Online].

[2] N. Perlroth. Hackers in China attacked The Times for last 4 months. New York Times, page A1, 31 January 2013.

[3] D. Gross. 50 million compromised in Evernote hack. CNN, 4 March 2013.

[4] D. Mirante and C. Justin,"Understanding Password Database Compromises",Dept. of Computer Science and engineering Polytechnic Inst. of NYU, Tech. Rep. TR-CSE-2013-02,2013.

[5] M.H. Almeshekah, E.H. Spafford, and MJ Atallah, "Improving Security using Deception", Center for Education and Research Information Assuarance and Security,Purdue University, Tech. Rep. CERIAS Tech Report 2013-13,2013.

[6] K.Brown,"The Dangers of Weak Hashes", SANS Institute InfoSec Reading Room, Tech. Rep., 2013.